SYSTEM AND METHOD FOR OPTIMIZING
STORAGE OPERATIONS

COPYRIGHT NOTICE

5   A portion of the disclosure of this patent document contains material which is subject to

copyright protection. The copyright owner has no objection to the facsimile reproduction by

anyone of the patent document or the patent disclosures, as it appears in the Patent and

Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

RELATED APPLICATIONS

10   This application claims priority to provisional application number 60/411,202 filed

September 16, 2002, the entirety of which is hereby incorporated by reference.

This application is related to the following pending applications, each of which is hereby

incorporated herein by reference in its entirety:

·   Application Serial No. 09/610,738, titled MODULAR BACKUP AND

15   RETRIEVAL SYSTEM USED IN CONJUNCTION WITH A STORAGE AREA NETWORK,

filed July 6, 2000, attorney docket number 044463-002;

·   Application Serial No. 09/609,977, titled MODULAR BACKUP AND

RETRIEVAL SYSTEM WITH AN INTEGRATED STORAGE AREA FILING SYSTEM, filed

July 5, 2000, attorney docket number 044463-0023;

20   ·   Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND

RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 044463-0014;

·   Application Serial No. 09/774,268 titled LOGICAL VIEW AND ACCESS TO

PHYSICAL STORAGE IN MODULAR DATA AND STORAGE MANAGEMENT SYSTEM,

filed, January 30, 2001, attorney docket number 044463-0035;

Application Serial No. 09/038,440, titled HIGH-SPEED DATA TRANSFER

MECHANISM, filed March 11, 1998, attorney docket number 044463-0002; and

Application Serial No. 10/303,640 titled SELECTIVE DATA REPLICATION

SYSTEM AND METHOD, filed November 25, 2002, attorney docket number 4982-21PROV.

5

BACKGROUND OF THE INVENTION

Field of the invention

The invention relates to data storage in a computer network and, more particularly, to a

system and method for optimizing storage operations.

10

Description of related art

The GALAXY data storage management system software manufactured by

COMMVAULT SYSTEMS, INC. of Oceanport, New Jersey, uses storage policies to direct how

data is to be stored. Referring to Fig. 1, there is shown a library storage system 100 in

15 accordance with the prior art. Storage policies 20 in a management server 21 may be used to

map copy data from a source 24, through a media agent 26 to a physical media location 28, 30,

32, 34, 36, 38 using e.g., tapes, drives, etc., where data is to be stored. Storage policies 20 are

generally created at the time of installation of each media library, and/or stand alone drive.

Numerous storage policies may be created and modified to meet storage management needs. A

20 storage policy allows the user to define how, where, and the duration for which data should be

stored without requiring intimate knowledge or understanding of the underlying storage

architecture and technology. The management details of the storage operations are transparent to

the user.

Storage policies 20 can be viewed as a logical concept that direct the creation of one or more copies of stored data with each copy being a self-contained unit of information. Each copy may contain data from multiple applications and from multiple clients or data sources. Within each copy are one or more archives, relating to a particular application. For example, one

5   archive might contain log files related to a data store and another archive in the same copy might contain the data store itself.

Storage systems often have various levels of storage. A primary copy or data set, for example, indicates the default destination of storage operations for a particular set of data that the storage policy relates to and is tied to a particular set of drives. These drives are addressed

10   independently of the library or media agent to which they are attached. In Fig. 1, the primary drives are media 28, 30, 32, 34, 36 and 38. Clearly other forms of storage media could be used such as tapes or optical media. The primary data set might, for example, contain data that is frequently accessed for a period of one to two weeks after it is stored. A storage administrator might find storing such data on a set of drives with fast access times preferable. On the other

15   hand, such fast drives are expensive and once the data is no longer accessed as frequently, the storage administrator might find it desirable to move and copy this data to an auxiliary or secondary copy data set on a less expensive tape library or other device with slower access times. Once the data from the primary data set is moved to the auxiliary data set, the data can be pruned from the primary data set freeing up drive space for new data. It is thus often desirable to

20   perform an auxiliary storage operation after a primary data set has been created. In Fig. 1, the auxiliary data set is copied to drives or tapes 40, 42 and 44.

Storage policies generally include a copy name, a data stream, and a media group. A primary copy name may be established by default whenever a storage policy for a particular

3

client is created and contains the data directed to the storage policy. A data stream is a channel

between the source of the data, such as data streams 50 and 52 in Fig. 1 and the storage media

such as data streams 50 and 52 in Fig. 1. Such a data stream is discussed in HIGH-SPEED

DATA TRANSFER MECHANISM, serial Number 09/038,440 referenced above. To increase

5   the speed of a copy, data to be backed-up is frequently divided into a plurality of smaller pieces

of data and these pieces are sent to a plurality of storage media using their own respective data

streams. In Fig. 1, data from source 24 is broken into two portions and sent using streams 50,52

to media 28, 36.

A client's data is thereby broken down into a plurality of sub-clients. In Fig. 1, media

10   28, 30, 32 and 34 may comprise a single media group and media 36 and 38 a second media

group. A media group generally refers to a collection of one or more physical pieces of storage

media. Only a single piece of media within the group is typically active at one time and data

streams are sent to that media until it achieves full capacity. For example, data stream 50 will

feed source data to medium 28 until it is full and then feed data to media 30. Multiple copies

15   may be performed using multiple streams each directed to a respective media group using

multiple storage policies.

Auxiliary copying, discussed in more detail in commonly owned Application Serial No.

10/303,640, denotes the creation of secondary copies, such as medium 40 or medium 42, of the

primary copy. Since auxiliary copying involves multiple storage policies and data streams which

20   each point to a particular media group, data is likely scattered over several pieces of media.

Even data related to single stream copy operations might also be scattered over several media.

Auxiliary copying is generally performed on a stream-by-stream basis and one stream at a time,

in an attempt to minimize the number of times the primary media are mounted/unmounted. For

example, for a copy of 10 pieces of primary media where four streams are used, auxiliary copying first entails copying all archive files of the first stream to a first set of auxiliary media, then the second stream to a second set of auxiliary media, etc. In Fig. 1, an auxiliary copy of stream 50 is made using auxiliary stream 50a to medium 40 and, if needed, medium 42.

5   Thereafter, an auxiliary copy of stream 52 is made using auxiliary stream 52a to medium 44.

An archive file, at least with respect to auxiliary copying, is generally copied from a first chunk of data to a last chunk. When an auxiliary copy operation is cancelled or suspended before all chunks of an archive file are successfully copied to the destination copy, the chunks successfully copied are generally discarded or overwritten later when the archive file is again

10   copied to the same copy or medium. This is undesirable because it wastes time and resources to copy the same chunks repeatedly; it wastes media because useless data occupies the media until the media is reusable; and if the network is not stable, a large archive file may never be successfully copied.

Although the GALAXY data storage management system software provides numerous

15   advantages over other data storage management systems, the process for restoring copied data may require access to several media, which involves multiple mounting/unmounting of media, thereby increasing the time necessary for a restoration. Additionally, although an effort is made to minimize the number of times media are mounted and unmounted, the stream-by-stream basis used in auxiliary copying does not minimize the number of mount/unmount times necessary for

20   the auxiliary copy and does not minimize tape usage. For example, in Fig. 1, media 40 and 44 may both be less then half full but both are needed to copy data through streams 50a, 52a using conventional techniques and both must be remounted for a restore. Performing auxiliary copying on a stream-by-stream basis is also generally a lengthy process. Finally, restarting a copy of an

archive file that has been cancelled or suspended by always copying the first to the last chunk is

inefficient with respect to media usage and the time necessary to complete a copy.

There is therefore a need in the art for a system and method for increasing the efficiency

of storage management systems.

5

SUMMARY OF THE INVENTION

A system and method for transferring data in a library storage system. The library

storage system comprises a media server including a storage policy. A media agent is connected

10   to the media server. A plurality of storage media and a data source are connected to the media

agent. The media agent divides the data source into at least a first and a second portion of data.

The portions of data are transferred from the data source to a first and second primary storage

medium using a first and a second data stream respectively. The media agent then causes the

first and second portion of data to be transferred from the first and second storage medium to a

15   third auxiliary storage medium using a third combined data stream. Auxiliary copying is

performed in chunks and multiple streams are copied in parallel.

One aspect of the invention is a method for transferring data in a library storage system.

The library storage system comprises a management server. A media agent is connected to the

management server. A plurality of storage media are connected to the media agent and a data

20   source is connected to the media agent. The method comprises dividing the data source into at

least a first and a second portion of data. The method further comprises transferring the first and

second portion of data from the data source to a first and second storage medium using a first and

a second data stream respectively. The method still further comprises transferring the first and

6

second portion of data from the first and second storage medium to a third storage medium using a third combined data stream.

Another aspect of the invention is a system for transferring data. The system comprises a data source, a media agent connected to the data source and a management server connected to

5 the media agent. The system further comprises at least a first, second, and third storage medium connected to the media agent. The data source is divided into at least a first and a second portion of data. The media agent transfers the first and the second portion of data from the data source to the first and second storage medium using a first and second data stream respectively. The media agent transfers the first and second portion of data from the first and second storage

10 medium to the third medium using a third combined data stream.

Still another aspect of the invention is a recording medium in a storage system with data stored thereon. The storage system comprises a management server, a media agent connected to the management server, a plurality of storage media connected to the media agent, and a data source connected to the media agent. The data is produced by splitting data source into at least a

15 first and a second portion; transferring the first portion to a first storage medium using a first stream; transferring the second portion to a second storage medium using a second stream; and transferring the first and second portion of data from the first and second storage medium to a third storage medium using a third data stream.

Yet still another aspect of the invention is a method for transferring data in a storage

20 system. The storage system comprises a management server, a media agent connected to the management server, a plurality of storage media connected to the media agent, and a data source connected to the media agent. The method comprises dividing the data source into at least a first and a second portion of data. The method further comprises transferring the first and second

7

portion of data from the data source to a first number of pieces of storage media. The method further comprises transferring the first and second portion of data from the first number of pieces of storage media to a second number of pieces of storage media, the second number being less than the first number.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the operation of a library storage system in accordance with the prior art.

10   Fig. 2 is a block diagram showing the operation of a library storage system in accordance with the invention.

Fig. 3 is a flow chart detailing some of the operations of an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

15

The efficiency of data storage management systems is increased in the invention by providing a system and method that combines data streams of one or more storage policies during an auxiliary copying operation. Combining data streams generally denotes copying or backing-up archive files associated with different streams, onto a single or a fewer number of

20   streams, thereby minimizing the number of media required for an auxiliary copy operation and consequently reducing the number of mount/unmount times necessary.

Combining streams may be enabled by allowing a plurality of applications in a source to be copied to point to a single storage policy. Referring to Fig. 2, there is shown a library storage

8

system 200 in accordance with the invention. Data from a plurality of applications (e.g.

EXCHANGE, WORD, EXCEL, etc.) from a source 25 is controlled via media agent 27

according to storage characteristics specified by a storage policy 19 in a management server 23.

The data are each copied to a respective medium, such as a hard drive, tape, etc through streams

5   54, 56, 58. For example, for a primary copy or storage of three applications, the data from each

of the applications will be saved to tapes 58, 60, 62, 64, 66 and 68, respectively, through streams

54, 56, and 58 respectively, thereby requiring at least three tapes for the copy operation, i.e.,

tapes 58, 62, and 66. During the auxiliary copy operation which combines streams, the data on

tapes 58, 62 and 66 are combined into fewer media – i.e., tape 70 and, only if needed, tape 72.

10   This may be accomplished, for example, by a storage policy pointing to either the same or

another media library for storing the auxiliary copies. Thus, the primary copy operation requires

three tapes, but the auxiliary copy is reduced to one tape, assuming the capacity of one tape is

sufficient to hold the data from the three applications.

The archive files may be given an application identification, e.g., appId (sub-client), and

15   the files are copied by default in ascending order according to the appId in order to minimize the

impact on restore speeds. Alternately, combining streams can be done stream-by-stream for the

fastest copying times as discussed below.

In addition to combining data streams, data is copied to the auxiliary media 70, 72 in a

logical order, such as in the order of the primary copy according to the date the primary archive

20   files were create. In this way, the archive files of a copy may be copied together in a single

medium, allowing users to more easily determine which medium contains a particular copy.

Combining streams helps in media recycling. For instance, assume that there are several

primary copies on four media that correspond to four streams and archive pruning has pruned all

but one copy. The remaining copy may still hold up the four media. If the primary copy is copied, job-by-job, into one stream to an auxiliary copy, the primary copy will be copied onto one medium and the other three primary media are then recyclable.

The option of combining data streams may be selected or specified as an optional copy

5   method at the time a particular storage policy is created or defined. The combined data stream copy method may be applied to either synchronous data replication or selective data replication. Primary copies requiring multiple streams will generally not be copied to a medium with copies using combined streams. A copy made pursuant to a storage policy that combines streams generally cannot be changed into a copy that doesn't combine streams and vice versa.

10   A storage policy that combines streams includes a property, which may be selected or defined, that may be used for specifying the order for which data will be copied to media, e.g., a copy order. By default the copy order is in the order of application and job (explained below). This enhances efficiency with respect to restoring data from the copy. Alternately, the user may specify that the data be copied in the order of the stream number which is more efficient but

15   yields a high penalty for restores.

The "order of application and job" technique works as follows. All copy jobs within a given instance/copySet are copied together, e.g., all jobs selected for each client, appType, and instance/copySet. The jobs, e.g., the archive files, are then copied in the ascending order of their archive file Identification ("Ids"). Once a job copy is started, all the job's archive files are

20   copied together, even if those numbers are higher than other archive file Ids.

For example, if a copy set has two sub-clients with following archive files: (1) Sub-client 1: Job 1, archive file1 ("AF1") created 2:00 pm, (2) Sub-client 2: Job 2, archive file 2 ("AF2") created 3:00 pm, (3) Sub-client 1: Job 1, archive file 3 ("AF3") created 4:00 pm (resumed and

<div align="center">10</div>

finished or a multi-archive file job like exchange database), and (4) Sub-client 1: Job 3, archive file 4 ("AF4") created 5:00 pm. The copy order is AF1, AF3 (to finish Archive files of the job being copied), AF2, and AF4.

When a property or feature of the primary copy is changed or modified, the copy order of

5   each auxiliary copy that combines streams may also be changed. For example, if the primary copy was copied on a non-magnetic medium and now will be copied on a magnetic medium, the copy order will automatically be set to in order of application and job for all secondary copies. Otherwise, there will be generally no change in the copy order for the secondary copies. After the primary copy has been changed, the former primary copy by default will not combine data

10   streams.

During the creation of a storage policy for a nonmagnetic media group or drive, the graphic user interface ("GUI") includes a form element, e.g., check box, that allows the user to select the combine data stream option. The option is preferably checked OFF by default. Users can select the option by selecting or turning the feature ON in the Copy Policy interface screen in

15   order to enable the combine data stream option.

If the combine data stream option is selected, the copy order property will be enabled which allows the user to select from one of two choices: in order of stream number and in order of application and job. For a storage policy or policies for auxiliary copies whose primary copy is saved or to be saved on magnetic media or drives, where the combine data stream option is

20   selected, the copy order is preferably in order of application and job by default. Otherwise, the default copy order is in order of application and job. The copy order can be changed from one to the other at any time.

The GUI may display a message, such as a popup message, in the following situations:

11

- Where the primary copy is stored or to be stored on non-magnetic media or drives, if the user selects the combine data stream option or changes the copy order, the GUI warns the user about a higher amount of mount/unmount and tape seek activity during restores that will occur if the combine stream option is in order of stream number or during

5   auxiliary copies if the option is in order of application and job.

- If the user tries to point a SQL or DB2 sub-client to a storage policy that has a copy with the combine data stream option selected, the GUI warns the user that the multi-stream SQL or DB2 copies will not be copied using combined streams.

- If a storage policy is pointed to by a SQL or DB2 sub-client and the user tries to create a

10   new copy with the combine data stream option selected or tries to select the combine data stream option for an existing copy, the GUI warns the user that the multi-stream SQL or DB2 copies will not be copied to an existing copy using combined streams.

An archive manager is a computer program or instruction that manages archive

15   operations, such as creating and updating a storage policy, and retrieving data related to a storage policy. The archive manager may be implemented as an application or module and resides on a reference storage manager or media agent. An archive manager is preferably embodied in an ArchiveManagerCS class that is implemented as an Application Program Interface (API). The class further interfaces with at least one database or table which preferably includes the details of

20   storage policies, such as the copy name, data stream, media group, combine data stream properties, etc. The database or table includes values such as streamNum and flags, which indicate the selection of the combine data stream option. Additionally, the database or table may be accessed by other object classes, which may use the relevant data contained therein.

12

The stream number of an archive file copy is passed to a createCopy() method included in CVArchive. Additionally, an AuxCopyMgr process sends the stream number of an archive file copy to a remote auxCopy process in a CVA_COPYAFILE_REQ message.

All copies associated with a storage policy have the same number of streams, e.g., the

5   maximum number of streams, of the storage policy. This does not mean that a library for each copy has to have the same number of drives. A primary copy needs enough drives to support multi-stream copy. An auxiliary copy that combines data streams actually needs only one drive for auxiliary copying and for data restoration. Consequently, the associated library can be a stand-alone drive. In order to take advantage of stream consolidation, users that select the

10   combine stream option are allowed to create a storage policy pointing to a storage library with fewer auxiliary drives than copies.

Backup and synthetic full backups are allowed, which include a backup process writing the streamNum related to a storage policy into the archFileCopy table rather than archFile table when each archive file is created. The archive manager preferably handles this process.

15   A file system-like restore (involving indexing) includes one or more sub-clients. The sub-client restorations, may be performed serially, one at a time, in an arbitrary order or based on archive file location. For example, for each sub-client restore, archive files may be restored chronologically, such as in the order that the files were created. Alternatively or in addition, files may be restored, according to their offsets, such as restoring in order of offsets ascending within

20   each archive file. Offset refers to the distance from a starting point, e.g., the start of a file. Movement within an archive file typically corresponds with higher physical offsets from the beginning of the archive file.

The archive files in a secondary or auxiliary copy that are created by combining data streams are by default ordered as required for restoration. Restore efficiency could therefore be better with the auxiliary copy than with the primary copy. With respect to combining data stream-by-stream, the order of the archive files on media holding an auxiliary copy, may not

5   agree with the order the primary copies were created, which may require backwards tape movement during the restore. Backwards tape movement, the need to rewind, may be correctly handled by programming, such as by DATAMOVER software by GALAXY, during data restoration. Backward movement, however, has a negative impact on performance. A multi-stream ORACLE or INFORMIX copy can be restored from a single stream. However,

10   backwards tape movement during the restore may occur.

It is preferred that a copy involving multiple streams will not be copied to a copy medium that combines streams. Single stream copies may be copied to a copy medium that combines streams.

Referring to Fig. 3, there is shown a summary of the operations of the invention with

15   respect to combining streams. At step S102, the storage policy is queried or the user is asked whether the combine streams option should be enabled in his copy. If the user answers no or the storage policy indicates no, control branches to step S112 and copying is performed as in the prior art. Otherwise, control branches to step S106, and the system determines whether the streams can be combined. For example, auxiliary copy of SQL data should be the same number

20   of streams as the primary copy. If the streams cannot be combined, control branches to step S104, the user is informed that the streams cannot be combined, and copying is performed as in the prior art in step S112.

14

If the streams can be combined, control branches to step S108 and the data is backed up to the primary storage media using a desired number of streams. Thereafter, control branches to step S110 where the auxiliary copy is performed combining data streams.

5   Copy Restartable at Chunk

As stated above, in prior art auxiliary copying systems, auxiliary file copying restarts from a first chunk if the auxiliary copying was interrupted. This means if the copying operation is stopped in the middle, all copied chunks need to be copied again.

10   In the invention, auxiliary copying is performed such that data chunks of an archive file that have been successfully copied to a copy medium are not discarded and the copy operation resumes copying where the previous copying left off; auxiliary copy operations are restartable by a chunk, as opposed to restartable at archive file. Copying that is restartable at a chunk may be achieved with an API which calls a class that includes methods that do not delete the copied

15   chunks. For example, the createArchFileCopy() method in the ArchiveManagerCS class may include an instruction so that the successfully copied chunks are not deleted. A method may further be included to retrieve the last chunk copied for each archive file to be copied, such as a getToBeCopiedAfilesByCopy() method in the ArchiveManagerCS class. Additionally, new fields may be added into the CVA_COPYAFILE_REQ, such as messagearchFileSeqNum,

20   startChunkNum, startLogicalOffset and startPhysicalOffset fields.

The process for restarting a copy at chunks includes the AuxCopyMgr process checking if the archive file to be copied has chunks that were successfully copied to the copy media. If chunks have been copied successfully, the AuxCopyMgr process retrieves variables archFileSeqNum, startChunkNum, startLogicalOffset and startPhysicalOffset for the archive file

15

and sends them to the AuxCopy process in the CVA_COPYAFILE_REQ message. For each

stream of the destination copy, the AuxCopyMgr process starts copying from the archive file that

has chunks that were successfully copied. The AuxCopy process calls CVArchive::createCopy()

using the parameters archFileSeqNum, startChunkNum, startLogicalOffset and

5   startPhysicalOffset. This allows AuxCopy to start writing or copying from the correct chunk and

offset. The AuxCopy process may also call DataMover::Seek() with startPhysicalOffset as one

of the input parameters to find the starting chunk and offset before the first DataMover::Read()

call.

Additionally, the CVArchive::createCopy() API, which is used by AuxCopy, includes

10   input parameters archFileSeqNum, startChunkNum, startLogicalOffset and startPhysicalOffset.

When startChunkNum > 1, the API does not send a CVA_CREATEAFILECOPY_REQ message

to commServer for creating an archFileopy entry since there is one already. The API also uses

the parameters passed in to it to call Pipelayer::create().

15   Multi-Stream Auxiliary Copy

In another aspect of the invention, methods and systems are provided which allow multi-

stream auxiliary copying. In the prior art, auxiliary copying is performed one stream at a time no

matter how many streams are used during a copy. The amount of time for copying a copy job is

therefore proportional to the number of streams used during a copy. This is referred to as single-

20   stream Auxiliary Copying.

In the invention, multi-stream Auxiliary Copying refers to performing auxiliary copies

for a plurality of streams in parallel. This may be accomplished by providing a sufficient

number of drives so that each stream may copy to at least one drive, thereby reducing the time

16

necessary for auxiliary copies involving multiple streams. For example, in an instance where two drives are required for each stream (e.g., one source and one destination), the number of streams that can be copied at the same time is half of the number of available drives. If six streams were used for copy jobs, an auxiliary copy job can copy archive files for three streams at

5   a time if there are six drives available, and can copy archive files for six streams at a time if there are twelve drives available, etc.

The process for multi-stream auxiliary copying includes the AuxCopyMgr process reserving more than one stream for the same destination copy or for multiple destination copies at same time. One stream is assigned to one destination copy at a time. If the AuxCopyMgr

10   process has not reserved enough streams, the process will keep trying if some streams are temporarily not available. When a copy is done with a stream for a destination copy, the AuxCopyMgr first releases the stream then tries to reserve the next stream (the copy can be different). The AuxCopy process is able to run more than one worker thread that copies an archive file for a stream and each thread uses its own pipeline. When an Auxiliary Copy job is

15   interrupted, stopped, or cancelled, the AuxCopy process stops all the worker threads and exits, and the AuxCopyMgr process releases all the streams and exits. If an AuxCopy process fails to copy for one stream, the worker thread reports the failure to AuxCopyMgr process and exits. The AuxCopy process continues to run until no work thread is running or is stopped by AuxCopyMgr. Depending on the nature of the failure, the AuxCopyMgr process decides

20   whether it is necessary to stop copying archive files for all streams of a copy or stop copying archive files for all copies.

Thus, by combining streams in auxiliary copying, auxiliary copy operations are optimized. By allowing auxiliary copies to be performed by chunk, auxiliary copying may be

17

performed more efficiently even if the copying is interrupted. By allowing for multiple stream

auxiliary copies, auxiliary copying may be performed even more quickly than that available in

the prior art.

Although the invention has been described in connection with the GALAXY data

5   management system by way of example, it is understood that the disclosure may be applied to

other data management systems, and references to the GALAXY system should therefore not be

viewed as limitations.

While the invention has been described and illustrated in connection with preferred

embodiments, many variations and modifications as will be evident to those skilled in this art

10   may be made without departing from the spirit and scope of the invention, and the invention is

thus not to be limited to the precise details of methodology or construction set forth above as

such variations and modification are intended to be included within the scope of the invention.